

Applicant : Matthew J. Adiletta et al.
Serial No. : 09/811,995
Filed : March 19, 2001
Page : 6 of 10

Attorney's Docket No.: 10559-320001 / P9681

REMARKS

Applicant amended independent claim 17 to include the feature recited in claim 22 that the one or more bytes loaded into the destination register are specified by a field of the instruction representing a mask. Applicant similarly amended independent claim 26. Applicant cancelled claims 22 and 27, and amended claims 23-24 and 28-29 to correct their dependencies.

After these amendments claims 17-21, 23-26 and 28-29 are pending in the above-referenced patent application. Claims 23 and 26 are independent.

The examiner rejected claims 17-29 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,663,012 to Shimizu et al., in view of the reference "Programming, Compiling and Executing Partially-Ordered Instruction Streams on Scalable Shared-Memory Multiprocessors" by D. K. Probst (hereinafter Probst).

The examiner also rejected claims 17-21 and 25-26 under 35 U.S.C. §103(a) over the reference Computer Systems Design and Architecture by V. P. Heuring and H. F. Jordan (hereinafter Heuring), in view of Probst. In addition, the examiner rejected claims 22-24

Applicant's independent claim 17 recites "a local register instruction that loads one or more bytes, specified by a field of the instruction representing a mask, within a local destination register with a shifted value of another operand." Applicant's instruction enables bytes of a destination register to be selectively loaded with source data. For example, as described in the originally filed application, "[the mask] 0101 loads the first and 3rd bytes [of the destination register] while the other bytes remain unchanged" (page 11, line 27).

With respect to the feature which appeared in old claim 22, and which pertained to applicant's use of a mask field in the instruction to specify which bytes of the destination register to load data into, the examiner stated:

11. Referring to claim 22, Shimizu has taught wherein the local register instruction comprises a field representing a mask that specifies which byte or bytes of the destination register are affected (Shimizu column 4, lines 12-34; column 4, line 51 to column 5, line 40; column 10, line 6 to column 11, line 47; Figure 7; Figure 8; Figure 9; Figure 9; Figure 16; and Figure 17) (Office Action, page 5, paragraph 11)

Applicant : Matthew J. Adiletta et al.
Serial No. : 09/811,995
Filed : March 19, 2001
Page : 7 of 10

Attorney's Docket No.: 10559-320001 / P9681

Applicant disagrees.

Shimizu describes a data processor and instruction that are capable of inserting and extracting data to and from optional bit area of a register, and to a control circuit therefor (col. 1, lines 11-15). Specifically, Shimizu describes two types of instructions; the GETxx instructions, and the PUTxx instructions. Every GETxx instruction, when executed on Shimizu's processor, causes a specific predetermined byte of the source operand to be placed into the lowest portion of a destination operand (FIG. 9, col. 4, line 49 to col. 5, line 14). For example, the instruction GETB0 extracts the first byte from the head of the source operand and places it in the lowest byte of the destination operand (FIG. 9A, FIG. 14, and col. 4, lines 53-57). In a similar vein, execution of one of Shimizu's various PUTxx instructions causes a portion of a source operand to be placed into a specific predetermined byte position of a destination operand (FIG. 10, col. 5, lines 15-40). For example, execution of the instruction PUTB0 causes the lowest byte of the source operand to be placed into the first byte (most significant byte) of the destination operand (FIG. 10A, FIG. 17, col. 5, lines 21-24, and col. 10, line 50 to col. 11, line 5).

However, nowhere does Shimizu describe any instruction that uses an instruction field, representing a mask, to specify which bytes of the destination operand to load data into. Rather, as explained above, to place a source data into a specific desired byte of the destination register, it is necessary to execute that GETxx or PUTxx instruction that would cause source data to be placed into that desired byte of the destination register. If it is desired to load data into a different byte(s) of the destination register, it is necessary to execute a different GETxx or PUTxx instruction to accomplish that.

Moreover, in discussing the format of its GETxx and PUTxx instructions, Shimizu explains:

FIG. 7 shows the formats of the GETxx instruction and the PUTxx instruction which are provided for the data processor of the invention.

The GETxx instruction is the one which extracts data from an optional byte position of a register. The PUTxx instruction is the one which inserts data at an optional byte position of the register.

In FIG. 7, numeral 401 designates the format of both instructions described above.

Applicant : Matthew J. Adiletta et al.
Serial No. : 09/811,995
Filed : March 19, 2001
Page : 8 of 10

Attorney's Docket No.: 10559-320001 / P9681

The format 401 of these instructions comprises, in order from the head byte, a first operation code part (1 byte) 4011, a source operand specifying part (1 byte) 4012, a source operand extension part (0 to 2 bytes) 4013, a second operation code part (1 byte) 4014, a destination operand specifying part (1 byte) 4015 and a destination operand extension part (0 to 2 bytes) 4016.

With the source operand specifying part 4012 and the destination operand specifying part 4015, such addressing modes of format as shown in the schematic drawing of FIG. 8 can be specified, with a 16-bit or 32-bit extension part being placed behind the operand specifying part depending on the addressing mode.

In FIG. 8, numeral 411 designates a format of a register direct addressing mode, numeral 412 designates a format of a register indirect addressing mode, numeral 413 designates a format of a 16-bit register relative indirect addressing mode, numeral 414 designates a format of a 32-bit register relative indirect addressing mode, numeral 415 designates a format of a 16-bit absolute addressing mode, and numeral 416 designates a format of a 32-bit absolute addressing mode.

The source operand of the GETxx instruction and the destination operand of the PUTxx instruction can specify only the register direct addressing mode, and the instructions are identified by the value of the second operation code part 4014. (col. 4, lines 11-48)

Thus, none of the fields of any of the GETxx and/or PUTxx instructions includes a field representing the mask that specifies one or more bytes into which data from a source operand is to be loaded.

Accordingly, Shimizu neither discloses nor suggest at least the feature of "a local register instruction that loads one or more bytes, specified by a field of the instruction representing a mask, within a local destination register with a shifted value of another operand," as required by applicant's independent claim 17.

With respect to the other references cited by the examiner, the examiner admitted that:

31. Regarding to claims 22-24 and 27-29, Heuring In view of Probst have not taught
 - a. Wherein the local register Instruction comprises a field representing a mask that specifies which byte or bytes of the destination register are affected (Applicant's claim 22);

... (Office Action, Page 11, Paragraph 31)

Applicant : Matthew J. Adiletta et al.
Serial No. : 09/811,995
Filed : March 19, 2001
Page : 9 of 10

Attorney's Docket No.: 10559-320001 / P9681

Thus, since none of Shimizu, Heuring, and Probst discloses or suggests, alone or in combination, at least the feature of "a local register instruction that loads one or more bytes, specified by a field of the instruction representing a mask, within a local destination register with a shifted value of another operand," applicant's independent claim 17 is therefore patentable over the cited art.

Claims 18-21, 23 and 24 depend from independent claim 17 and are therefore patentable for at least the same reasons as claim 17.

Independent claim 26 describes an apparatus featuring "a command that causes the ALU to load one or more bytes, specified by a field of the command representing a mask, within a destination register of a selected microengine with a shifted value of another one or more bytes of a source register." For similar reasons as those provided with respect to independent claim 17, at least this feature is not disclosed by the art. Claim 28-29, which depends from claim 26, are patentable for at least the same reasons as claim 26.

It is believed that all the rejections and/or objections raised by the examiner have been addressed.

All of the dependent claims are patentable for at least the reasons for which the claims on which they depend are patentable.

Canceled claims, if any, have been canceled without prejudice or disclaimer.

Any circumstance in which the applicant has (a) addressed certain comments of the examiner does not mean that the applicant concedes other comments of the examiner, (b) made arguments for the patentability of some claims does not mean that there are not other good reasons for patentability of those claims and other claims, or (c) amended or canceled a claim does not mean that the applicant concedes any of the examiner's positions with respect to that claim or other claims.

FEB. 27. 2006 2:06PM

(3) FISH & RICHARDSON 6175428906

NO. 8189 P. 11

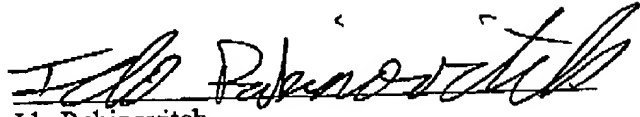
Applicant : Matthew J. Adiletta et al.
Serial No. : 09/811,995
Filed : March 19, 2001
Page : 10 of 10

Attorney's Docket No.: 10559-320001 / P9681

No fee is believed due. Please apply any charges or credits to deposit account 06-1050,
referencing attorney docket 10559-320001.

Respectfully submitted,

Date: Feb. 27, 2006


Ido Rabinovitch
Attorney for Intel Corporation
Reg. No. L0080

PTO Customer No. 20985
Fish & Richardson P.C.
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

21274390.doc